

# **Quarterly Progress Report**

**N01-NS-1-2333**

## ***Restoration of Hand and Arm Function by Functional Neuromuscular Stimulation***

**Period covered: April 1, 2004 to June 30, 2004**

**Principal Investigator:** Robert F. Kirsch, Ph.D.

**Co-Investigators:**

Patrick E. Crago, Ph.D.  
P. Hunter Peckham, Ph.D.  
Warren M. Grill, Ph.D.  
J. Thomas Mortimer, Ph.D.  
Kevin L. Kilgore, Ph.D.  
Michael W. Keith, M.D.  
David L. Wilson, Ph.D.

Dawn Taylor, Ph.D.  
Joseph M. Mansour, Ph.D.  
Jeffrey L. Duerk, Ph.D.  
Wyatt S. Newman, Ph.D.  
Harry Hoyen, M.D.  
John Chae, M.D.  
Dustin Tyler, Ph.D.

**Program Manager:** William D. Memberg, M.S.

Case Western Reserve University  
Wickenden 407  
10900 Euclid Avenue  
Cleveland, OH 44106-7207  
216-368-3158 (voice)  
216-368-4969 (FAX)  
rfk3@po.cwru.edu

## Contract abstract

The overall goal of this contract is to provide virtually all individuals with a cervical level spinal cord injury, regardless of injury level and extent, with the opportunity to gain additional useful function through the use of FNS and complementary surgical techniques. Specifically, we will expand our applications to include individuals with high tetraplegia (C1-C4), low tetraplegia (C7), and incomplete injuries. We will also extend and enhance the performance provided to the existing C5-C6 group by using improved electrode technology for some muscles and by combining several upper extremity functions into a single neuroprosthesis. The new technologies that we will develop and implement in this proposal are: the use of nerve cuffs for complete activation in high tetraplegia, the use of current steering in nerve cuffs, imaging-based assessment of maximum muscle forces, denervation, and volume activated by electrodes, multiple degree-of-freedom control, the use of dual implants, new neurotization surgeries for the reversal of denervation, new muscle transfer surgeries for high tetraplegia, and an improved forward dynamic model of the shoulder and elbow. During this contract period, all proposed neuroprostheses will come to fruition as clinically deployed and fully evaluated demonstrations.

## Summary of activities during this reporting period

The following activities are described in this report:

- *Musculo-skeletal modelling of the shoulder and elbow*
- *Development of a 3D virtual reality environment for testing command sources*
- *Evaluation of Command Interfaces for a High Tetraplegia Neuroprosthesis*
- *Wireless data acquisition module for use with a neuroprosthesis.*
- *Rapid prototyping and real-time control for functional electrical stimulation (FES)*

## Musculo-skeletal modelling of the shoulder and elbow

**Contract section:** E.1.a.ii Model-based development of neuroprostheses for high tetraplegia

### Abstract

A large-scale, three dimensional model of the upper extremity that combines the advantages of forward and inverse dynamics is under development. The model uses inverse-dynamic optimization to calculate an optimal set of neural excitations for a given movement. These excitations can be used as the starting point for investigations using a forward-dynamic model, which would otherwise be prohibitively expensive in terms of computational cost. Initial results show that the algorithm is successful in providing muscle excitations which produce the desired movement.

### Introduction

The application of biomechanical modelling can helpfully be divided into two categories: fundamental research into human function, and analysis of clinical problems. Much insight can be gained into human function by the use of simple models that illustrate certain principles of movement or motor control. When applied to the evaluation of specific clinical problems,

however, a large-scale three-dimensional model is essential. Only such a comprehensive model allows for the validation that is essential to clinical application, and the ability to make patient-specific predictions of clinical outcome. This allows sophisticated models to be used for the analysis of clinical problems, improvement of current treatments and the development of new procedures. This work focuses on the further development of such a model that will allow its use in the solution of clinical problems.

## Methods

The Delft Shoulder and Elbow Model (DSEM) as first described by van der Helm (1994) is an inverse-dynamic model of the complete shoulder mechanism. The recorded motions of the bones are used as inputs to the model, together with the external loading, and internal forces and moments are calculated using an inverse-dynamics method. The musculo-skeletal system is indeterminate; that is, there are many more muscles crossing a degree of freedom than equations of motion governing it. Thus, some kind of optimization must be carried out to calculate individual muscle forces. The 'load-sharing problem' was solved by a non-linear optimization routine involving the minimization of the sum of the squared muscle stresses. Muscle force-length relationships are also accounted for by the inclusion of muscle architecture parameters. Output from the model consists of the resulting joint contact forces, ligament forces and muscle forces, lengths and moment arms.

The model has been used by a number of authors for the study of activities such as wheelchair propulsion (van der Helm and Veeger 1996, Veeger et al. 2002), the examination of surgical interventions such as tendon transfer (Magermans et al. 2004A, Magermans et al. 2004B) and the treatment of scapular fracture (Chadwick et al. (*in press*)), and for the design of a neuroprosthetic system by Kirsch et al. (2001).

The motion of the upper limb is subject to a number of constraints which complicate the modelling process. The most obvious of these is the closed-chain nature of the shoulder girdle. That is, the thorax, scapula and clavicle form a closed-chain kinematic mechanism which restricts the motion of the shoulder. In modelling terms, this must be accounted for by a reduction in the number of degrees of freedom, and leads to a mismatch in the kinematics observed in a subject and that which is shown by the model. This can be accounted for by scaling of the bones in the model, or by adjustment of the input data. A further reduction in the freedom of the shoulder girdle is assumed to be caused by the conoid ligament, which couples the rotations of the clavicle and the scapula due to its high stiffness.

The inverse-dynamic (ID) model is quasi-dynamic in that segment velocities and accelerations are accounted for, but muscle dynamics are not. The equations of motion can thus be solved independently for each time-step, and so the model is computationally efficient. This does not allow studies to be made of the effects of muscle dynamics on function, however, and the post-operative kinematics resulting from a given intervention must be known *a priori*. When analyzing an existing normal or pathological motion this is fine, but for speculative studies about the effects of certain interventions, it is a serious limitation. In this, the advantages of a forward-dynamic model, in which the input muscle activations result in output motions, are clear.

Forward-dynamic (FD) models have the great advantage that no *a priori* knowledge of kinematic is necessary, but are less common in the literature for one reason in particular: computational cost. The optimization of the neural excitations which produce a desired limb trajectory involve the repeated integration of the state equations at each step, which leads to enormous simulation times for large-scale, three-dimensional models. One further advantage that

FD models have over their ID counterparts is the ability to model stiff structures such as ligaments. These structures are often of such stiffness that the accuracy required of displacement estimates to ensure realistic force changes is unachievable.

In order to take advantage of the possibilities offered by forward-dynamic models, a 'hybrid' model is under development. This model combines inverse and forward dynamics to allow FD simulations to be carried out in a feasible time frame. The model employs static optimization of muscle forces (inverse dynamics) together with feedback and feedforward control to drive an FD model towards some desired trajectory. The ID model is used to calculate optimum muscle forces which produce the desired motion, and an inverse muscle model calculates muscle excitations. These excitations are then used to drive an FD model, and the error in predicted kinematics is fed back to the ID model as a controller. This produces an optimum set of muscle excitations for a given motion, which can be used as a starting point for FD studies.

## Results

Simulations of simple motions such as abduction and anteflexion of the humerus have been successfully carried out, with approximately two hours needed to simulate one second of movement. While this is still slow, it is much faster than a full-scale forward-dynamic optimization would be on a model of this scale. The controller was able to produce a set muscle excitations which led to the correct motion being produced by the FD model. Errors between the kinematics of the ID and FD models were of the order of 2°. However, without the controller, the FD model is not stable, and discretization errors and other numerical instabilities are catastrophic.

## Next quarter

The next step in the development of the model is to achieve gains in speed. This can be done with the implementation of a faster integrator and a simpler muscle model, and these possibilities will be investigated. Furthermore, evaluation of the model needs to be done by comparison of the predicted muscle excitations with measured EMG signals. Refinement of the predicted excitations can then be done by modification of the cost function used in the muscle force optimization.

## References

- Chadwick, EK, van Noort, A and van der Helm, FC (2004) Biomechanical analysis of scapular neck malunion - a simulation study. *Clinical Biomechanics (in press)*.
- Kirsch RF, Acosta AM, van der Helm FC, Rotteveel RJ, Cash LA. (2001) Model-based development of neuroprostheses for restoring proximal arm function. *J Rehabil Res Dev*. 38 (6), 619-26.
- Magermans DJ, Chadwick EK, Veeger HE, Rozing PM, van der Helm FC. (2004A) Effectiveness of tendon transfers for massive rotator cuff tears: a simulation study. *Clin. Biomech*. 19 (2), 116-22.

Magermans DJ, Chadwick EK, Veeger HE, van der Helm FC, Rozing PM. (2004B) Biomechanical analysis of tendon transfers for massive rotator cuff tears. *Clin Biomech* 19 (4), 350-7.

van der Helm, F.C.T. (1994) A finite element musculo-skeletal model of the shoulder mechanism. *J. Biomech.* 27 (5), 551-69.

van der Helm, FC and Veeger, HE (1996) Quasi-static analysis of muscle forces in the shoulder mechanism during wheelchair propulsion. *J Biomech.* 29 (1), 39-52.

Veeger HE, Rozendaal LA, van der Helm FC. (2002) Load on the shoulder in low intensity wheelchair propulsion. *Clin Biomech* 17 (3), 211-8.

## **Development of a 3D virtual reality environment for testing command sources; Utilization of this environment to collect baseline data and evaluate brain signals as a command source.**

**Contract section:** E.1.a.iv Command Sources for High Tetraplegia

### **Introduction**

For individuals with high-level spinal cord injury (C4 and above), command sources are needed that will convey enough dimensions of control to restore reach and grasp within a reasonable workspace. These command sources are limited to the signals that can be volitionally produced from the neck up. To provide reach capabilities anywhere within a three dimensional (3D) workspace, a minimum of three proportional command signals are needed to convey wrist location within the 3D space. Additional proportional and discrete command signals will be needed to convey hand orientation, degree of finger flexion/extension, thumb position, and a hand lock command. This section reports on the progress in developing a generic virtual reality testing environment that will allow us to test various sources of command signals for eventual control of the full arm and hand FES system. This section also reports on our progress in utilizing brain recordings as a possible command source for this patient population.

### **Virtual Reality Testing Environment Update**

The virtual reality (VR) interface for three dimensional command source evaluation and training is near completion. The Matlab code has been written that presents targets in a 3D workspace and presents a moving cursor that the user must control with whatever command sources are being evaluated. This specific configuration of the VR system is specifically intended to test command sources for 3D control of wrist position in space. The code can later be expanded to include control of a virtual hand representation, or control of the full musculoskeletal model of an FES-activated limb.

At this initial testing stage, the code is configured to take in wrist position that is collected in real time through the Optotrak motion capture system. Once the final bugs are worked out with the stereo viewing of the virtual image, target-directed reaching movements

from healthy individuals will be collected to provide baseline performance statistics with which to compare the same target-directed movements when using various command sources to drive the cursor. Many of the data analysis functions have been written that will quantify various aspects of the virtual movements:

- trajectory variability
- directness of path
- uniformity of control throughout the workspace
- independence of each dimension
- information transfer rate
- stability during hold
- goal success rate

### **EEG-based command source update**

Work is also progressing on the development of a system that could in the future utilize cortical signals from scalp or brain surface EEGs, or from intracortical signals. The software that will use these neural signals as a command source in the virtual reality task is being finalized. The active-X code to get the EEG data into Matlab and analyzed for spectral content in real time is near completion but still has a few bugs to work out regarding buffer overloads. Matlab code has been written that will process these neural signals and translate them into virtual cursor movements using an adaptive decoder. This adaptive decoder is designed to iteratively improve itself based on the movement errors seen in recently attempted movements. This will allow the decoder to track learning induced changes in the volitionally produced brain signals. This code is expandable into any number of dimensions. Therefore, it will still work when virtual hand control is added on top of the current 3D virtual cursor position control.

### **Next Quarter**

The final bugs need to be worked out of each of these three components: 1) the virtual reality testing environment, 2) the real-time data collection and processing, and 3) the adaptive neural decoder. These three components will then be integrated together and utilized in the approved protocols for testing brain signals as a command source for continuous multidimensional control. Subjects will start with one dimensional control of the virtual cursor, then expand to two and then three or more dimensions of control as their skill level improves.

## **Evaluation of Command Interfaces for a High Tetraplegia Neuroprosthesis**

**Contract section:** E.1.a.iv Command Sources for High Tetraplegia

### **Abstract**

After a high cervical spinal cord injury, the amount of function that must be restored to replace lost hand and arm capability is substantial, while the number of voluntary actions available to serve as command inputs is markedly limited. The problem comes down to one of controlling the position and orientation of the hand in space with the remaining voluntary actions, and these are limited to the head, neck, and face due to the injury level. Despite the

significant loss of many voluntary actions, a number of potential command sources remain by which the user can command the eight degrees of freedom necessary to restore function. This study is an investigation of these potential command interfaces and their applicability as inputs to a neuroprosthesis for high tetraplegia. Quantitative measures of command interface performance are evaluated using a robot arm of similar dimensions to a human arm. The robot was programmed to simulate a paralyzed arm, such that potential command sources were tested as they would be applied to an FES device driving a paralyzed arm. Evaluation of the information passed by each of the possible command sources allows for comparison via a common quantity. The information transfer rate (ITR) of various command sources was measured using a modified three-dimensional Fitts' Law task.

### Methods:

Like a regular Fitts' Law task, the experiment involves having the subject move from a starting location to a target position. In this experiment, however, the subject instructs a robot arm to move instead of their own arm. The robot arm used is a small industrial assembly robot, mounted in an inverted manner and placed next to the subject's shoulder (Figure 1). This serves as a proxy for a paralyzed arm activated using FES, allowing the system to approximate a user with a neuroprosthesis. This approach also allows for comparison with conventional Fitts' Law experiments.

Three command interfaces were explored in a set of preliminary experiments. "Direct servo control", where the robot shoulder angles were mapped directly to head pitch and yaw, was employed as an intuitive, though attention-demanding interface. A constant velocity "gated ramp" control algorithm based on head pitch and yaw was the second interface tested. The third interface tested was a constant velocity gated ramp control using EMG signals from facial muscles as inputs.

Direct servo control slaves the motion of the arm to the subject's head orientation. A small orientation sensor worn on the back of the head (Figure 1b) measures the pitch, roll, and yaw of the subject's head. Of these, only pitch and yaw were used to control left/right and up/down motion of the robot arm. The net result was that the pointer mounted on the end of the robot points in the same direction as the subject's nose.

Head based gated ramp uses thresholds on either side of a baseline head orientation. When the user's head orientation deviates from the centerline beyond a specified amount, the arm moves in that direction at a constant speed. Nominal thresholds were  $\pm 20^\circ$  left and right,  $30^\circ$  up and  $-10^\circ$  down. These values can be changed based on user comfort.

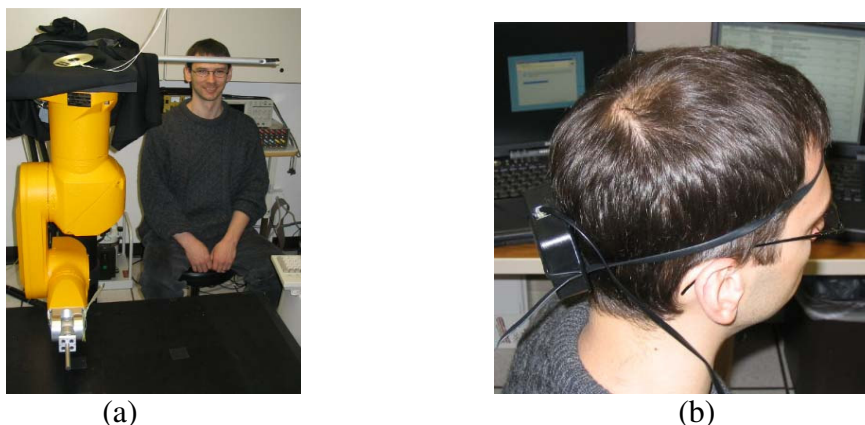
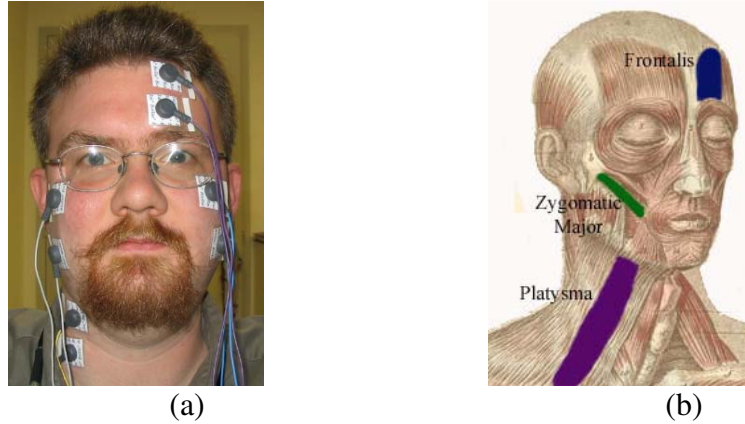


Figure 1. Photographs of a) subject position relative to robot, and b) head.

The third command interface explored used electromyogram (EMG) signals from four major muscle groups on the face (Figure 2). These signals were recorded, and muscle activity above a pre-set level activated a gated ramp algorithm to move the arm at a constant speed. Left and right motions were commanded using the corresponding zygomatic major muscle. Up commands were indicated with the frontalis muscle while the platysma was used for down instructions. Both of these later command sources were side independent.



**Figure 2. Electrode placement for facial EMG command interface illustrating a) placement on subject, and b) muscles used as command sources.**

The experiment consisted of having 9 healthy, normal subjects command the action of the arm using one of the three command interfaces previously discussed. Each command interface was tested by three subjects, with no cross-over of subjects between interfaces. Six target locations were used with 4 different target sizes, with the order of each randomly selected prior to the start of the experiment. The targets were spheres with diameters of 60mm suspended from the ceiling to various locations in the workspace of the robotic arm. The workspace of the arm is approximately the same as that available to an individual seated at a table.

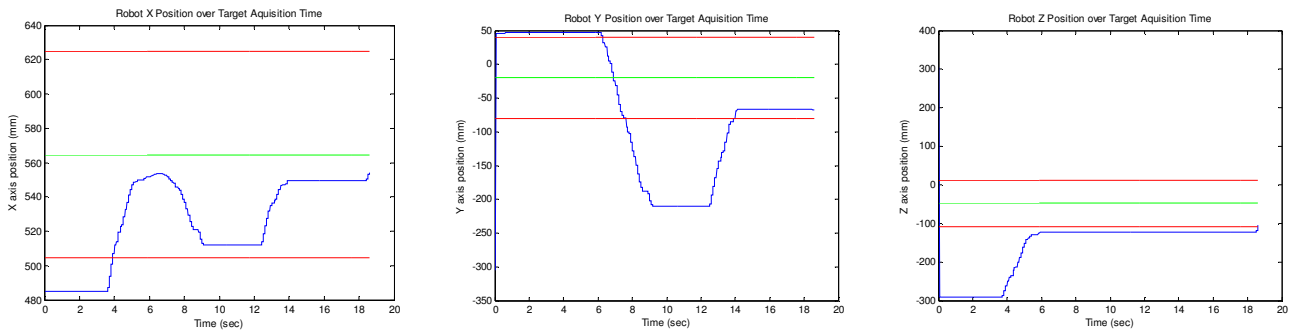
During the trial, the time to reach the target was recorded, as was the trajectory of the arm's endpoint. Based on the size of the target, its distance from the starting position, and the time required to reach the target, the information transfer rate (ITR) for that command interface was calculated using the following formula:

$$ITR = [\log_2(2A/W)]/t_m$$

Where A is the distance to the target, W is the radius and  $t_m$  is the movement time [4]. The units are Bits/sec.

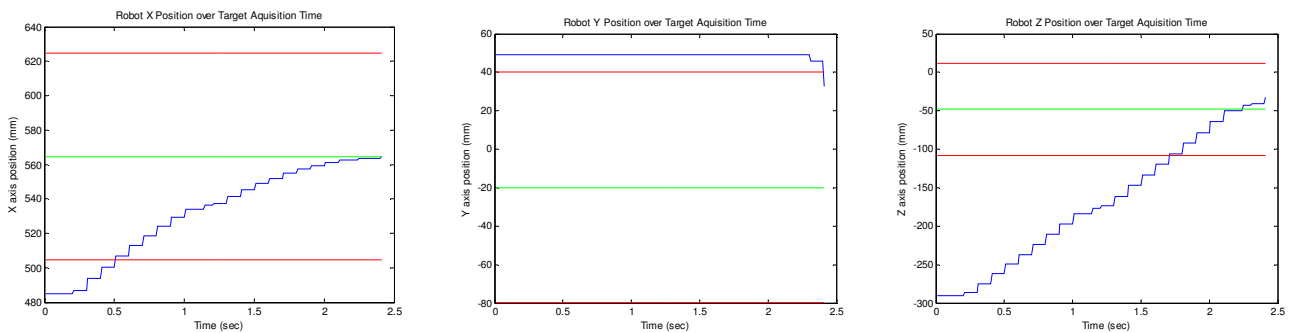
## Results

Figure 3 illustrates the trajectory of the endpoint of the arm pointer, from the home position (as seen in Figure 1a) to a point approximately at shoulder level at the center of the subject's chest, as commanded with head orientation-based gated ramp. This figure is an example of poor performance as the subject oscillated about the horizontal center (XY plane) of the target, leaving the approach to the vertical target to the end of the trial once the other two dimensions were "close" to the center of the target. This resulted in the subject taking a large amount of time to complete the task, in addition to not moving directly to the target as would be seen in a natural reaching movement.



**Figure 3. X, Y, and Z trajectories of an example of poor performance (0.16 B/s).**

Figure 4, however, illustrates exceptional performance, with direct motion in the coronal plane (XZ) to the same target as Figure 3, with final lateral motion at the end of the trial. This direct motion, without oscillation allowed for approximately 10 times faster movement time. This trial was conducted by a different subject using the facial EMG command source.



**Figure 4. X, Y, and Z trajectories of an example of good performance (1.3 B/s).**

Typical results from these preliminary experiments are presented in Table 1. The average ITR for servo drive and head based gated ramp were both 0.49 Bits/sec. EMG based gated ramp had an ITR of 0.55 Bits/sec. The results in Table 1 represent the average across all six targets for all three subjects for each command interface.

	Servo	Gated Ramp	EMG Gated Ramp
<b>Average (Bits/sec)</b>	0.49	0.49	0.55
<b>Std Dev (Bits/sec)</b>	0.29	0.22	0.28

**Table 1. Average and standard deviation of command source performance.**

## Discussion

Initial results using the above methods were favorable and correlate with the subjects' subjective evaluations regarding the perceived ease of use of the interfaces studied. Direct servo drive, while intuitively easy to use, required constant operator attention and some of the more extreme positions required placing the head in correspondingly extreme orientations. Head based gated ramp control showed comparable performance to servo drive, but subject feedback indicated that it was much more comfortable. EMG based commands showed the best performance, with subjects indicating that being able to maintain eye contact with the arm's endpoint (not always possible with the other interfaces) allowed for superior performance. Some problems with this method occurred however during more animated conversation, specifically laughing, which resulted in accidental commands.

Not surprisingly, the best performance was seen when straight-line trajectories were generated. Subjects who had a difficult time with the interface (as seen in Figure 3) had a significantly lower ITR. This is mostly a factor of training, illustrating the need for adequate time to become familiar with the command interface. It should be noted that these results were collected with a maximum of two minutes of training time in order to roughly gauge the effects of training upon user performance. As rough pass results, they illustrate the significant effect of user training upon the final ITR calculations.

## Next Quarter

Based on the results above, command interfaces using the orientation and muscle activity of the head neck and face can be both used to control a neuroprosthesis, and their performance can be compared using the information transfer rate. In the next quarter we will commence formal testing of these command interfaces. Long-term goals include testing of other command interfaces such as the RobotEyes object recognition and planning system from Spatial Integrated Systems, Inc., as well as determination of head orientation based on muscle activity of the neck. The final outcome of the study is the selection of a command interface for control of an implanted neuroprosthesis for restoration of hand and arm function.

## References

1. Agarwal, S., Triolo, R.J., Kobetic, R., Miller, M., Bieri, C., Kukke, S., Rohde, L., Davis, J.A. "Long-term user perceptions of an implanted neuroprosthesis for exercise, standing, and transfers after spinal cord injury". Journal of Rehabilitation Research and Development. Vol. 40, No. 3. May/June 2003. pp. 241-252

2. Acosta, A.M., Kirsch, R.F., Parikh, P.P. "Neuroprosthesis for restoration of shoulder function in C5-C6 tetraplegia". Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings. Vol. 5, 1997. pp. 1942-1943
3. Shannon, C.E. "A Mathematical Theory of Communication". Bell System Technical Journal. Vol. 27, July-Oct 1948. pp. 623-656
4. Wickens, C.D., Gordon, S.E., Liu, Y. "An Introduction to Human Factors Engineering". Addison-Wesley, ©1998
5. Murata, A., Iwase, H. "Extending Fitts' law to a three-dimensional pointing task". Human Movement Science Vol. 20, 2001 pp.791-805

## Wireless Data Acquisition Module for Use with a Neuroprosthesis

**Contract section:** E.1.a.v      Sensory feedback of contact and grasp force

### Abstract

A general wireless data acquisition module (WDAM) is being developed for use with a neuroprosthesis. The WDAM is intended to be used with sensors such as the shoulder or wrist position transducer, finger-mounted joysticks, or remote on-off switches. Currently these sensors are connected to a controller via cables, which are cosmetically unappealing to the user and often get caught on wheelchairs, causing them to be damaged. Switch-activated transmitters mounted on walkers have been used previously in FES applications [1]. Recent advances in wireless technology have reduced the complexity and size of the wireless circuitry and have increased the likelihood that a small, low power, reliable wireless link could be assembled from commercially available components.

### Methods

In the previous quarter, a printed circuit board (PCB) version of the WDAM was fabricated. The PCB version has the transceiver, microcontroller, power conditioning, and analog input conditioning sections of the circuit all placed on a single board (Figure 5). Initial tests had shown that 99.8% of the data packets were successfully received and acknowledged although half of them needed to be sent more than once before they were accepted.

### Antenna Type

In this quarter, comparisons were made of the transmissions success rates using three different types of antennas (Figure 6). A 916.5 MHz signal utilizes a 32.7 cm wavelength. It is common practice in RF applications to use a quarter-wavelength whip antenna over a ground plane. This requires an 8.2 cm wire length. A shorter whip antenna can be used if a section of the wire is coiled to form an inductor. The standard short-whip quarter-wave antenna shown in Figure 6 is easily fabricated, and with a 5.0 cm length, is appropriate for development tests. The compact quarter-wave antenna (Linx Technologies, JJB-series) utilizes a precision helical coil to create a tuned antenna that is only 1.7 cm long, making it possible to conceal the antenna inside a small package. If the compact quarter-wave antenna cannot be used in a package due to height

limitations, an alternative is to use a trace on the PCB with an appropriate inductor. The PCB version of the WDAM was designed with a one-tenth wavelength antenna trace that is used if the inductor is placed between the trace and the transceiver.

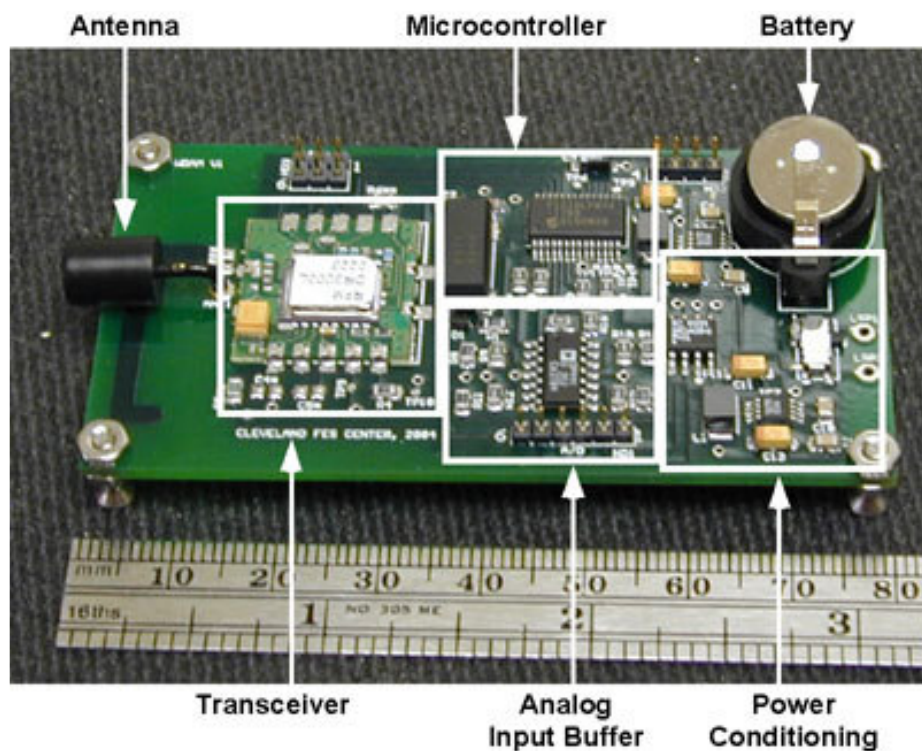


Figure 5. Printed Circuit Board version of Wireless Data Acquisition Module.

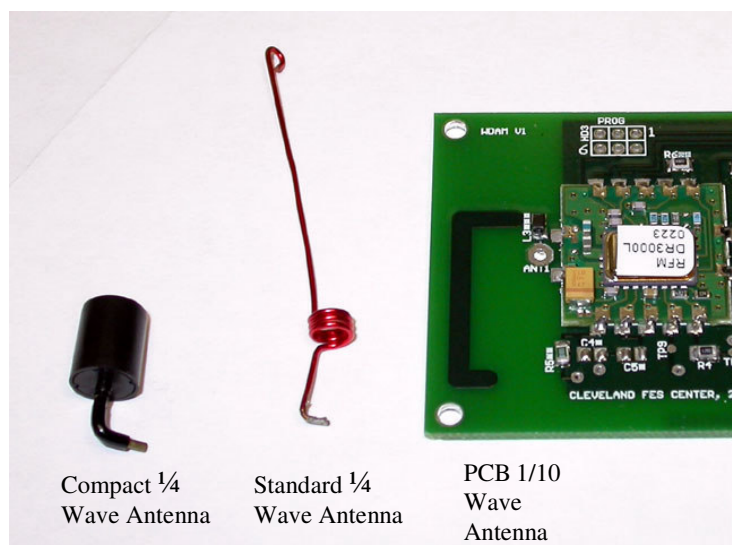


Figure 6. Antenna options for WDAM.

### Transceiver transmission speed

RF Monolithics produces three different versions of the DR3000 transceiver module. The WDAM board that was fabricated and tested last quarter used the “low-speed” version,

which is capable of data transmission rates of 2.4-19.2 kbps. The “medium-speed” version is designed for data transmission rates of 115.2 kbps. The “high-speed” version is designed for data transmission rates of 1 Mbps. In this quarter, two additional WDAM boards were fabricated. One incorporated the medium-speed transceiver, and one used the high-speed version. Tests of the transmission success rates were performed with each of these boards.

#### Wireless joystick application

In order to demonstrate an application of the WDAM, a force-sensing-resistor (FSR) joystick (Interlink Electronics, see Figure 7), which is similar to those used in some laptop computers, was connected to the four analog input channels of the “slave” WDAM. The “slave” WDAM was placed in an enclosure and the joystick was attached to the outside of the enclosure, allowing the “slave” WDAM and joystick to be held with one hand. The “slave” WDAM sent data to the “master” WDAM when it was requested. Since FSRs are nonlinear, the embedded software in the “slave” WDAM used a threshold measurement for each channel to determine which direction the joystick was being pushed, and sent a code for that direction to the “master” WDAM. The “master” WDAM then sent that data via a serial port to a computer running a simple LabView (National Instruments) routine that displayed which direction the joystick was being pushed (Figure 8).

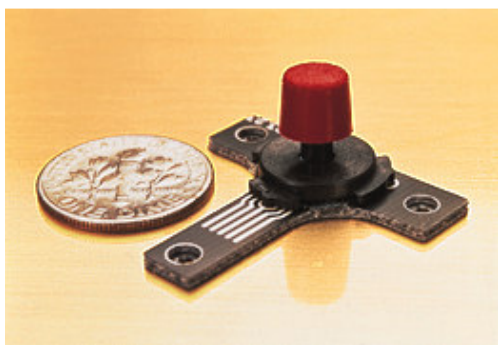


Figure 7. FSR-based micro-joystick.

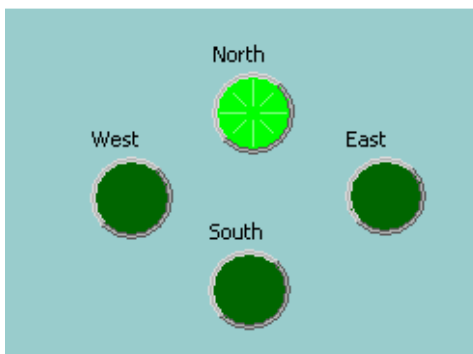


Figure 8. Simple directional output of LabView program for wireless joystick application.

## **Results**

### Antenna Type

All three antenna types performed similarly. When 1000 data packets were sent, 97-99% of the packets were successfully received and acknowledged, although half of them needed to be sent more than once before they were accepted. Therefore, the selection of an antenna will depend more on the application and the available space in the package.

These tests were performed with a five-foot separation between the transceivers. It is expected that the different antenna types will have different operating ranges and radiation patterns. This will be investigated more in the next quarter.

### Transceiver transmission speed

Both the “medium-speed” and “high-speed” transceiver versions had much higher receiver sensitivity than the “low-speed” transceiver version. This resulted in much more interference in packet transmission due to electrical noise. This noise often prevented any

packets from being transmitted successfully. A probable cause for these transmission failures is that the embedded software and filter settings had been optimized for the “low-speed” transceiver. More appropriate data packet parameters and filter settings will be evaluated in the future.

#### Wireless joystick application

The joystick connected to the “low-speed” WDAM provided a successful demonstration of a wireless application. Over 99% of the data packets were transmitted successfully during the tests. The graphical display clearly displayed the direction that the joystick was being pushed without any noticeable delay.

#### **Next Quarter**

In the next quarter, the “low-speed” WDAM will be further characterized. Tests will be performed on the operating range and orientation sensitivity. Other sensors will be tested, and the battery longevity of the WDAM with these sensors will also be evaluated. An additional WDAM board will be fabricated to allow the testing of multiple “slave” modules with one “master” module.

Also in the next quarter, we will investigate optimizing the data packet parameters and filter settings for the “medium-speed” and “high-speed” WDAM versions.

#### **References**

1. Z. Matjacic, M. Munih, T. Bajd, A. Kralj, H. Benko, and P. Obreza, "Wireless control of functional electrical stimulation systems," *Artif Organs*, vol. 21, pp. 197-200, 1997.

### **Rapid prototyping and real-time control for functional electrical stimulation (FES)**

**Contract section:** E.1.a.vi Implementation and Evaluation of Neuroprosthesis for High Tetraplegia

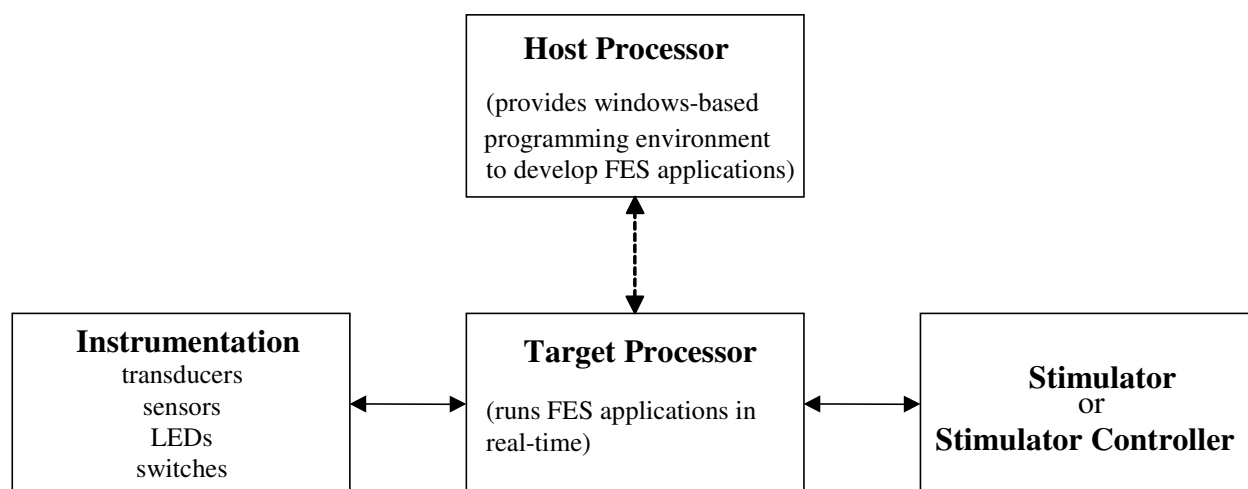
#### **Abstract**

In this project, we are developing a set of software and hardware tools to provide rapid prototyping and real-time control for FES systems. The abilities to read physiological data, process and make calculations based on those inputs, then adjust and communicate stimulation parameters at repeatable, well-defined time intervals are essential to control and develop algorithms for neuroprostheses. Typically, only computer engineers or programmers using C and/or assembly language have achieved such real-time control. This moves the development effort away from the researchers and/or clinicians that are directly studying the system. By using the much higher level and more abstract means of programming required for software packages from The MathWorks, Inc. (Natick, MA), we aim to put into the hands of these users the abilities to quickly design, implement, test, revise and re-test ideas in real-time. Our tools will provide a general mechanism for studying FES-related problems, and development of application-specific

solutions using these tools will be left to individual researchers or clinicians. Therefore, we will be able to centralize the technology development and support for the core software and hardware that serve many projects in this contract. During this quarter we have completed fabrication and have begun testing of the enhanced Prometheus Development Kit for use in a lab environment.

## Background

A typical FES system is shown in Figure 9. The "Host Processor" provides a windows-based programming environment to develop FES applications. The "Target Processor" runs the FES applications in real-time. It also provides the interface(s) for various inputs and outputs. In order to provide rapid prototyping and real-time control, we use personal computers (PCs) running MathWorks software for the "Host" (MATLAB/Simulink) and "Target" (xPC Target) processors, as described in Quarterly Progress Report #2 (QPR#2, Jul-Sept 2001). The "Stimulator Controller" is our portable, wearable Universal External Control Unit (UECU) that is serially connected to the "Target Processor". The UECU provides the interface to various stimulators/telemeters. This includes percutaneous electrodes and the implantable stimulator with myoelectric signal acquisition used in this contract.



**Figure 9. Basic Structure of a Functional Electrical Simulation System. (The Host Processor may be present only initially to download the application, or it may always be required to control the Target Processor.)**

## Methods

We are developing a MathWorks xPC Target compatible, PC based, Wearable Data Acquisition and Control System (WDACS) that utilizes our Universal External Control Unit (UECU) as the interface to various stimulators/telemeters. This WDACS is comprised of a commercial off the shelf (COTS), industry standard, PC/104 single-board computer (SBC), an in-house designed multifunction board, and user accessible batteries housed in a custom enclosure.

The Prometheus, a highly integrated PC/104 SBC with data acquisition (model PR-Z32-EA-ST, Diamond Systems Corporation, [www.diamondsystems.com](http://www.diamondsystems.com)) has been selected for the core of the system. The Prometheus SBC is realized on a single, industry standard, PC/104

printed circuit board, which measures 9 x 9.6 cm. Features of the Prometheus are summarized in Table 2. In particular, the Prometheus provides an extensive general-purpose data acquisition and control interface that provides for rapid configuration of various sensors, transducers, stimulators, and actuators. It is also supported with drivers for xPC Target.

The proposed work consist of two major phases; enhancement of the Prometheus Development Kit for use with the MathWorks xPC Target tools in a lab environment, and design and fabrication of the Wearable Data Acquisition and Control System. Experience with the enhanced development kit will help to finalize the design of the wearable system.

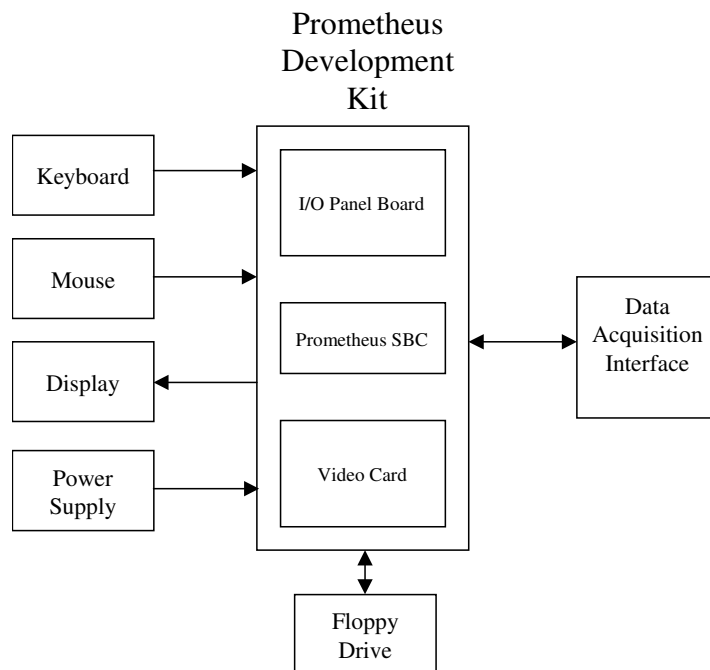
A Diamond Systems Prometheus Development Kit (CPR-Z32-EA-DK) was purchased to evaluate the functionality of the Prometheus SBC. The Prometheus Development Kit (PDK) provides everything needed to configure the Prometheus SBC with a keyboard, mouse, and display, and is housed in a 5.5 x 5.7 x 3.0 inch enclosure. A single 5V power supply provides all the necessary power. The kit also provides all of the connectors for interfacing with various peripheral devices.

**Table 2. Features of the Prometheus Single Board Computer.**

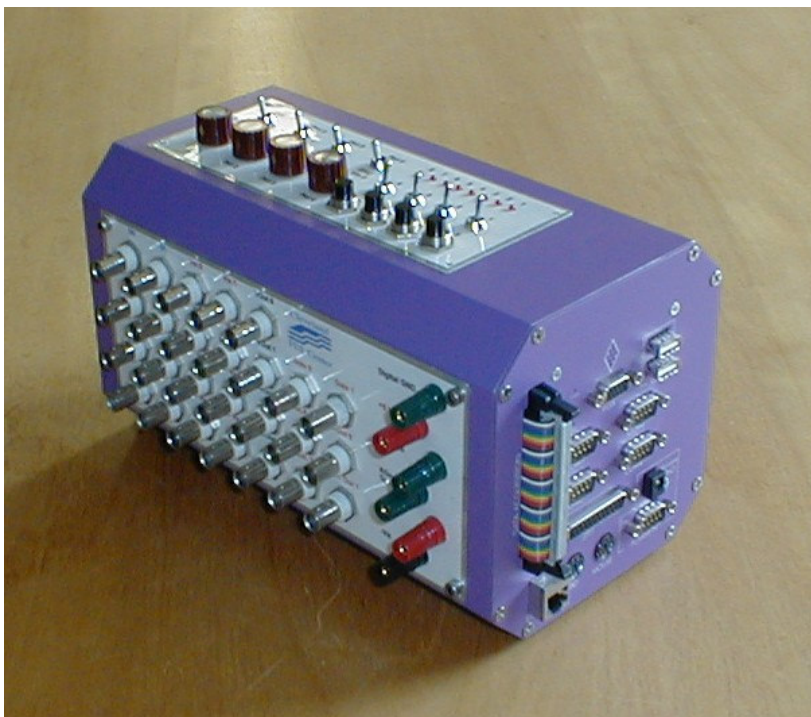
Pentium Class 486-DX2 Processor running at 100MHz with Co-Processor
32MB SDRAM System Memory
32MB Solid State FlashDisk
4 Serial Ports
2 Powered USB Ports
100BaseT Ethernet (100Mbps)
16 Channel, 16-bit Resolution, 100KHz (aggregate) ADC
Programmable Input Ranges/Gains
4 Channel, 12-bit DAC
24 Programmable Digital I/O
2 Programmable Timers
Programmable Gate and Count Enable

## Results

The PDK has been enhanced to include a floppy drive for initial loading of the xPC Target kernel and applications. The system has also been enhanced with a data acquisition interface for laboratory use. A block diagram of the Enhanced PDK is shown in Figure 10. A photograph of the Enhanced PDK is shown in Figure 11.



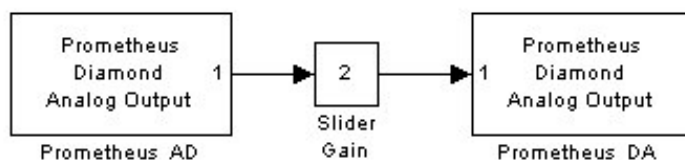
**Figure 10. Enhanced Prometheus Development Kit.**



**Figure 11. Photograph of the Enhanced Prometheus Development Kit.**

The data acquisition interface was fabricated to allow connection of lab-based instrumentation to the PDK. This interface provides BNC connectors for connecting to the ADC, DAC, counter/timer control, and digital input/output. In addition, the interface houses 4 potentiometers, 4 toggle switches, 4 pushbutton switches, and 8 LEDs. These general-purpose interfaces allow rapid hardware prototyping, complementing the rapid software prototyping provided by the MathWorks tools. Physically, the PDK, the floppy drive, and the data acquisition interface are housed in a single 25.5 x 14.5 x 14 cm enclosure as shown in Figure 11 above.

We have tested the functionality of the Enhanced PDK using MATLAB and Simulink on a desktop PC as the "Host" and xPC Target on the PDK as the "Target". A serial interface connects the Host and Target. The Target is booted from a floppy disk with the xPC Target real-time kernel. The rapid prototyping consists of five basic steps. First a model is created on the Host using Simulink or Simulink/Stateflow. This consists of creating a block diagram using a drag-and-drop graphical interface. Next the model can be simulated in nonreal-time on the Host to test for proper operation. The third step involves creating an executable target application using Real-Time Workshop, Stateflow Coder, xPC Target, and a C compiler. The fourth step is to execute the target application in real-time on the Target. Finally you can acquire signal data using scopes and gauge blocks and tune parameters from the Host.



**Figure 12. Simulink Block Diagram for the A/D D/A Test.**

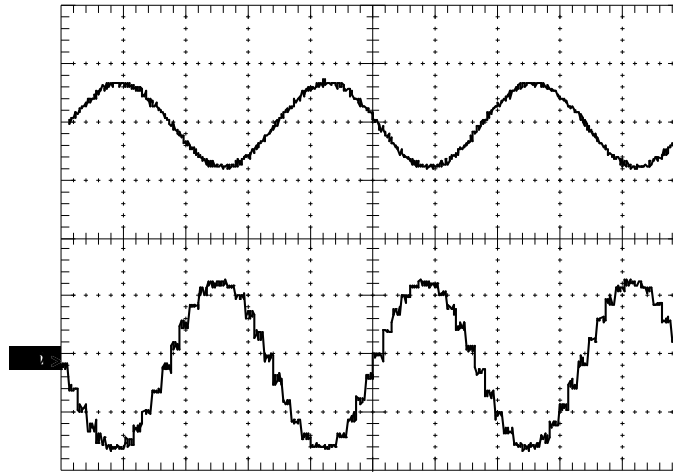


Figure 13. Input and Output Waveforms for the A/D D/A Test.

As a first test, a simple algorithm was implemented to read data from an A/D channel, multiply it by a slider-controlled constant (-2 to +2), and output the result on a D/A channel. The input waveform was a 300Hz, 7.5 V peak-to-peak sine wave. The sampling period was 150  $\mu$ s (6.6 KHz). The gain slider was -2. The Simulink block diagram used for the test is shown in Figure 12. Waveforms were measured on an oscilloscope are presented in Figure 13. The top trace is the input and the bottom trace is the output. Additional algorithms were developed to test the potentiometers, toggle and pushbutton switches, and LEDs.

### Next Quarter

We will deploy the Enhanced Prometheus Development Kit into the field. Experience with the use of the PDK will help to refine the appropriate functionality and configuration of the Wearable Data Acquisition and Control System. We will investigate the xPC Target embedded option for operating the PDK independent of the Host system. This will be necessary for operation of the portable system. We will also continue development of circuitry for the WDACS.